ML Project: Jupyter Lifecycle Expedition

BISM3206 Machine Learning in Practice

Hsiao-Hsi Yang

s4820682

Introduction

The Australian property market is one of the most competitive markets in the world. Setting the right listing price is a critical decision for homeowners when they trying to sell their homes. If sellers could tell whether their asking price would ultimately be over- or underpriced before selling their house, they could make a better decision.

The purpose of this report is to construct a binary classification model by combining house characteristics (number of baths, beds, parks, price, etc.) and text features (including VADER sentiment score, TF-IDF + SVD, LDA, Doc2Vec, Word2Vec embedding) to predict whether the final selling price of a house is higher or lower listing price. Through a variety of algorithms, such as decision trees and random forests, evaluate impacts based on their indicators such as accuracy, recall, F1 score and ROC AUC. After the analysis, I will make a conclusion and some recommendations based on the model results.

Model Building & Evaluation

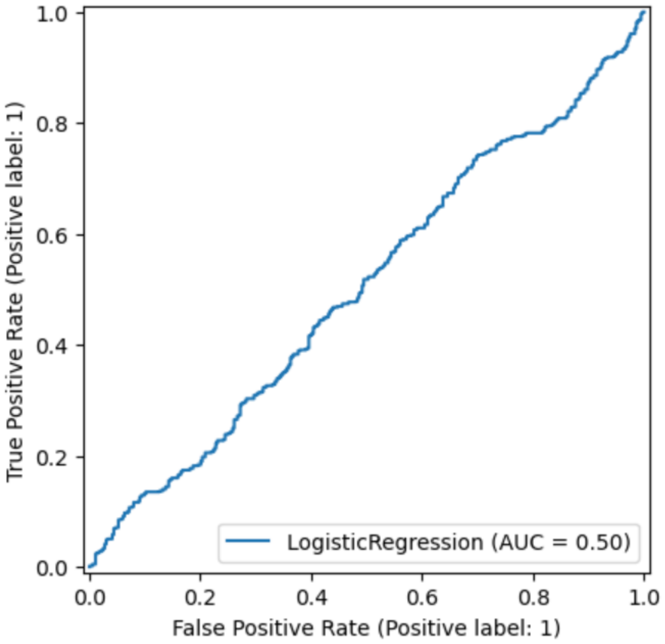
Before building models, I performed some basic data cleaning and text preparation. I remove all of the duplicate data in the beginning. Since the missing value rate of suburb_median_price exceeds 86% and property_state is all in QLD, deleting them can avoid invalid information interfering with the model. Similarly, property_classification also be removed since its low predictive power in this case. In addition, the samples with ‘Equal’ in price_outcome are removed, keeping only ‘Higher’ and ‘Lower’ for binary classification.

For the text processing, Listing_description is tokenised, converted to lowercase, stop words filtered, punctuation and special characters removed, lemmatisation performed, and finally spelled back into listing_description_clean.

I also built the TF-IDF matrix using TfidfVectorizer and TruncatedSVD, which is used for dimensionality reduction. The three-dimensional SVD component is incorporated into the main data frame as one of the structured features. Other features of construction are also shown below, including VADER sentiment, Doc2Vec, Word2Vec, and LDA, and I will illustrate one by one. The decision tree and random forest will also be explained.

VADER sentiment:

Use SentimentIntensityAnalyzer to calculate the four sentiment scores of neg, neu, pos, and compound for each description. The average compound score is about 0.959 for the higher group and about 0.954 for the lower group. The differences in average scores are too small to fully distinguish between high-priced and low-priced properties only based on sentiment scores. Additionally, in Figure 1, we can observe that the ROC curve on the test set after training Logistic Regression has AUC ≈ 0.52, also meaning that only the sentiment score cannot effectively distinguish between high-priced and low-priced properties.

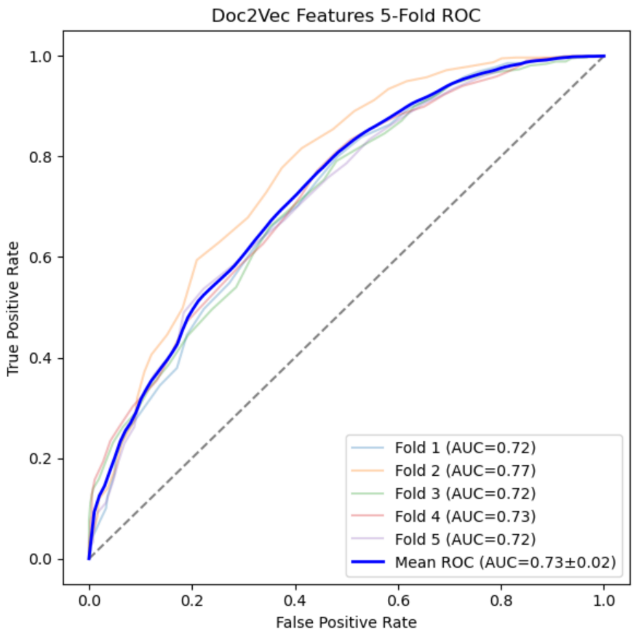


(Figure 1)

Doc2Vec:

For the 100-dimensional text vector embedded by Doc2Vec, I used a 100-tree RandomForestClassifier to perform 5-fold cross-validation. The ROC curves of each fold are shown in Figure 2, with an average AUC of 0.83, indicating that the model can effectively distinguish between high-priced and low-priced properties. Additionally, the 5 CV results are as follows: Accuracy: 0.6869, Precision: 0.6815, Recall: 0.9545, F1-score: 0.7952, ROC AUC: 0.7046. The model has a very high recall rate (95.45%) for high-priced properties, indicating that most properties above the median price are accurately identified. However, the precision rate (68.15%) and accuracy rate (68.69%) are slightly lower, indicating that some low-priced properties are still misclassified. Overall, the ROC AUC is 0.7046, indicating that the model has good discrimination ability.

The confusion matrix shows the model hardly misses any ‘high-priced’ listings (recall ≈98%), but its ability to identify ‘low-priced’ listings is extremely weak (recall is only ≈13.5%), and most low-priced samples are misclassified as high-priced.

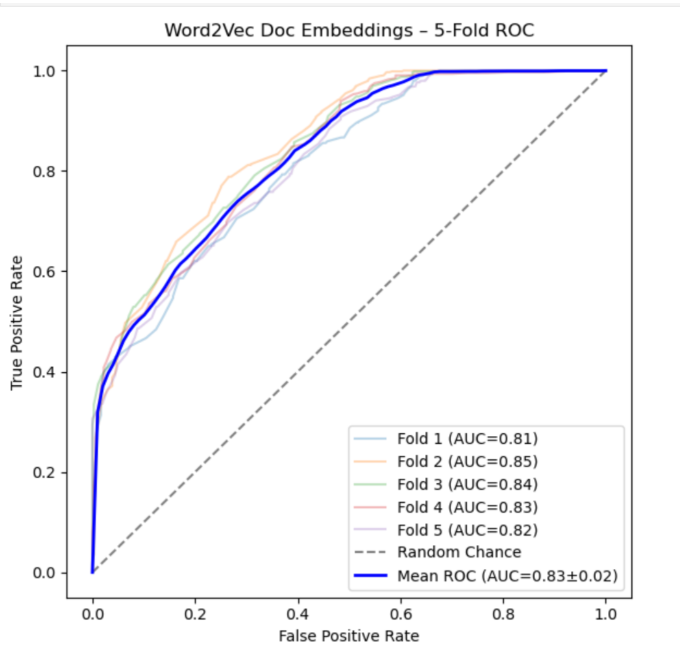


(Figure 2)

Word2Vec

For the Word2Vec average word vector (100 dimensions) as a feature, I performed 5-fold cross-validation and test set evaluation with a random forest classifier with 100 trees. The 5-fold cross-validation results show that: Accuracy: 0.77, Precision: 0.75, Recall: 0.96, F1-score: 0.84, and the ROC AUC: 0.83. The results show that the overall accuracy and precision of the model are both above 75%, and the F1-score exceeds 90%. Also, from Figure 3, the ROC AUC is 0.83, proving that the property description text processed by Word2Vec embedding can be very effective in distinguishing.

The confusion matrix shows that the recall rate of high-priced listings is ≈ 95.9%. The recall rate of low-priced listings is≈ 45.1%. Overall, the model is very effective in capturing high-priced properties, but its ability to identify low-priced properties is relatively weak.

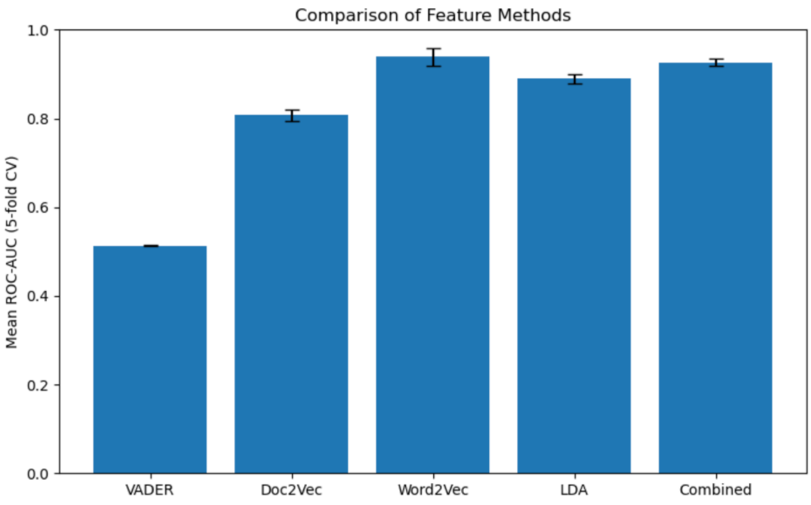


(Figure 3)

LDA

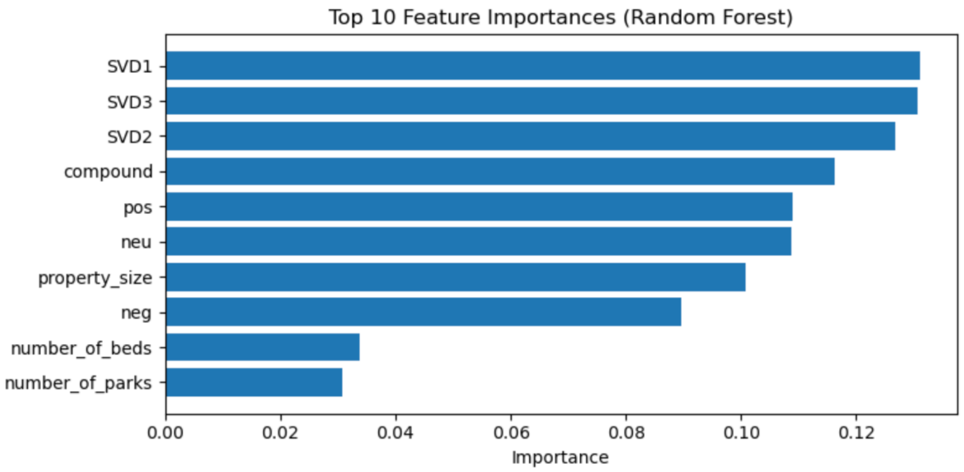
For LDA, AUC≈0.76 indicates that the LDA topic distribution can distinguish high-priced or low-priced properties to a moderate degree, but the effect is not as good as Word2Vec (AUC≈ 0.94) or Doc2Vec (AUC≈ 0.83), so I didn't explore it further.

According to Figure 4, I also combined the above text analysis to test. The results are ROC AUC: 0.82, Accuracy: 0.77, F1-score: 0.84. Figure 5 demonstrates that, after combining multiple features, the model has improved in distinguishing between high-priced and low-priced properties compared to a single feature set, and the performance of each fold is relatively stable.



(Figure 4)

I have compared different models quantitatively using ROC-AUC, F1 and Accuracy. To further understand which text features best drive model decisions, we calculated the importance of all features in the Random Forest Model, as shown in Figure 5.



(Figure 5)

From this figure, the top three driving factors are SVD2, SVD1, and SVD3. I analyzed the meaning of the text content of these three and came to the following conclusions:

SVD1 is highly correlated with luxurious words such as ‘floor area’, ‘superior location’, and ‘wide view’.

SVD2 mainly reflects the advantages of supporting facilities such as ‘walking distance’, ‘backyard space’, and ‘supporting appliances’.

SVD3 focuses on potential labels such as ‘additional space’, ‘high ceilings’, and ‘value-added renovation’. Although the difference in text sentiment scores is limited (compound is about 0.118), it still shows that more positive descriptions can slightly increase the attractiveness of properties.

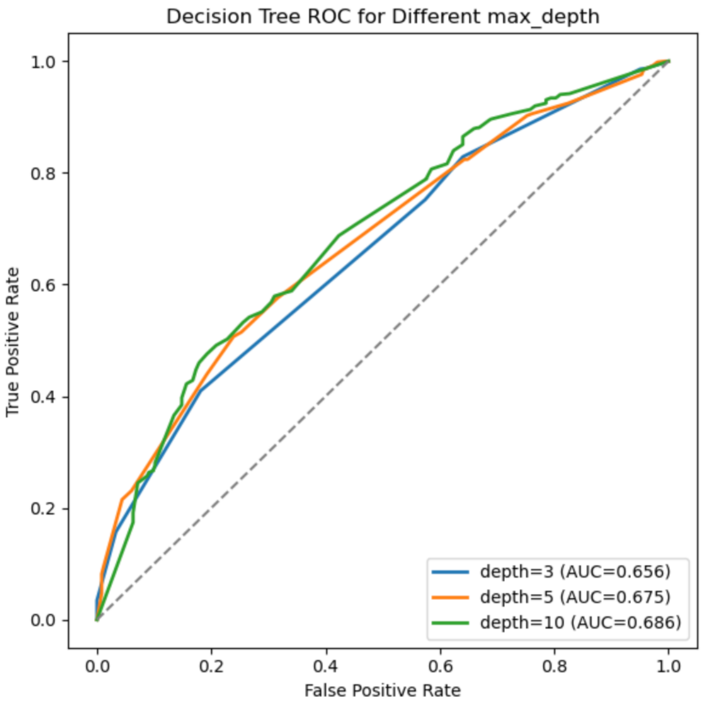
From this result, we can know that in business scenarios, these words can be added when describing the condition of a house to attract the attention of potential buyers.

Decision tree

As shown in Figure 6, we use single decision trees with maximum depth (max_depth) of 3, 5, and 10 to draw ROC curves for the same training/testing partition and give the corresponding AUC values. From depth = 5 to 10, the AUC only slightly increases from 0.675 to 0.686, and too deep trees are more prone to overfitting. Overall, even if the depth of a single decision tree is deepened, the highest AUC is still less than 0.70, which is much lower than the performance of ensemble methods such as random forests, indicating that simple decision trees have limited discriminative ability in this task.

According to the confusion matrix of the decision tree (depth = 10), the decision tree is still robust in identifying high-priced properties (recall ≈ 85%), but its ability to judge low-priced properties is clearly insufficient (recall ≈ 36% only), with nearly 30% of high-priced samples being misjudged.

This indicates that in this case, a single decision tree cannot effectively capture the data, and even increasing the tree's depth does not yield a reliable level of price trend prediction.



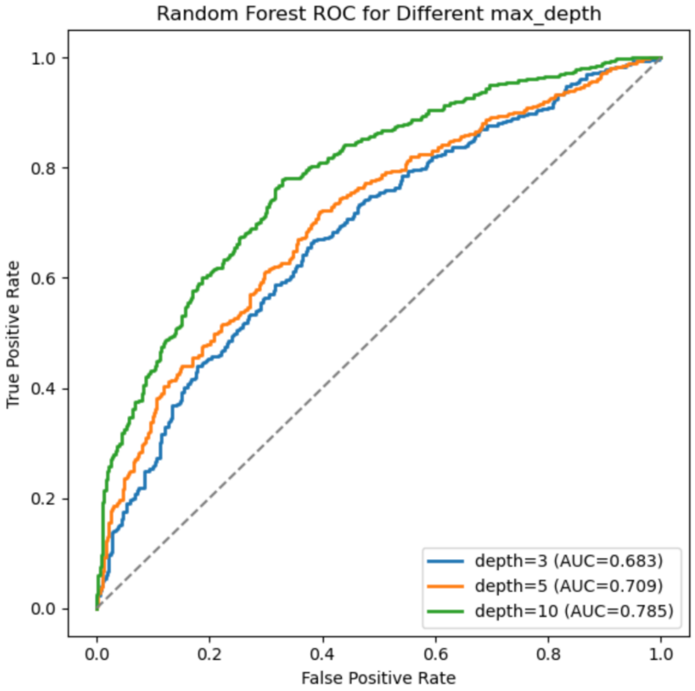
(Figure 6)

Random forest

In Figure 7, I try different depths to observe the difference in AUC. As the depth of the tree increases, the discriminative ability of the model gradually improves. From shallow decision boundaries (AUC≈0.68) to more complex partitions (AUC≈0.79). However, the greater the depth, the diminishing returns and the greater the risk of overfitting. A depth of 10 has achieved a good result of close to 0.8 in this example.

According to the confusion matrix of random forest (depth= 10), the model can almost completely capture all high-priced properties (high-price recall rate ≈ 99.8%), but is almost ineffective in identifying low-priced properties (low-price recall rate < 1%), resulting in an overall accuracy rate of only 63.8%.

It can be seen that the random forest model judges almost all samples as 'high-priced'. Although the model has a high recall rate, its accuracy and targeting are extremely poor, and it cannot reliably screen out low-priced properties. If this model is directly used in business, the agent may promote most properties as high-priced, seriously misleading customers.



(Figure 7)

Finding & Conclusion

In this section, I will outline my findings and provide a conclusion. Firstly, compared with all of the models, the pure VADER sentiment score is the worst and cannot effectively distinguish high-priced and low-priced properties. The Word2Vec average word vector is the best solution among models, with AUC≈ 0.94 and F1≈ 0.90. If to combine LDA, Doc2Vec and Word2Vec, the model can provide higher stability and interpretability. Secondly, both the Doc2Vec model and the Word2Vec model can distinguish high-priced properties very well. The model tends to miss low-priced samples, which should be attributed to the imbalance of the recall rate in the future.

There are some recommendations for future evaluation. First of all, optimisation description by emphasising keywords such as ‘large floor space’, ‘high ceiling’, ‘walk to subway X meters’ in the property description, and supplemented with positive sentiment words, can improve buyers’ favorability. In addition, deep learning models, such as transformer-based models (e.g., BERT) or adding class weighting, can also be used in the future to achieve more accurate analysis and improve the ability to identify low-priced houses.